大致流程

移动自动化的测试的路线：编程语言基础--->测试框架--->android/IOS开发测试基础---->appium API ----->开发移动自动化项目。

准备工作——软件安装

安装Appium（附带需要安装node.js和npm ） 用于自动化测试

安装JDK, Android SDK

安装模拟器，打开开发者选项（夜神）

1. 准备工作——adb命令学习（用于测试调试）
2. 查看版本 adb version
3. 连接模拟器 adb connect 127.0.0.1:62001
4. 进入shell adb shell

多设备时指定设备： adb -s 端口号 shell

Eg: adb -s 127.0.0.1:62001 shell

退出shell exit

1. 在设备上安装apk
   1. 直接把.apk拖入模拟器
   2. 命令行： adb install “apk路径”

可选参数：-r 覆盖原安装文件 -s 指定设备

如果failure,考虑进shell,chmod 777 local然后重装

1. 卸载apk

Step1: adb shell

Step2: cd /data/app/

Step3: adb uninstall <apk名>

可选参数： -k 卸载软件，但保留配置和缓存文件

1. 查看设备上安装的应用包名：

adb shell

pm list package

可选： | grep “需要过滤的包名”

1. 文件读写

7.1 将文件从PC写入设备

adb push <remote路径（电脑路径）> <local路径（虚拟机/真机路径）>

7.2 将文件从设备读到PC(不能直接pull到电脑磁盘跟目录)

Adb pull <remote> <local>

1. 屏幕截图 adb shell screencap /保存路径/截图名.后缀（用png jpg有兼容性问题）
2. adb服务启动和关闭：

adb kill-server

Adb start-server

1. 解除端口占用

netstat -ano | findstr “xxx（端口号）”

Taskkill -f -pid xxx

Tip:可将常用的adb命令封装成bat命令，方便运行，不用每次手动输入

封装建议使用Notepad 避免编码问题

1. Package和Activity
2. Package

关于包名：app的包名唯一

* 1. 打开UI autoviewer
  2. adb shell pm list package可获取所有包名

1. Activity

关于Activity：一般相当于手机屏幕的一个页面

* 1. Activity获取——aapt命令

2.1.1 aapt dump badging “apk路径” 一般用于获取

launchable-activity

注：可用 aapt dump badging “apk路径” | findstr “launchable-activity”进行过滤

* 1. Activity页面布局元素

FrameLayout LinearLayout RelativeLayout AbsoluteLayout TableLayout TextView ImageView

（感觉不是很重要，所以不详细描述了，类似前端页面的容器和布局）

1. 测试工具——Monkey

用于进行程序压力测试，通过生成伪随机的用户事件流，检测和评估被测程序的稳定性,且会对测试系统进行监测

5.1 Monkey路径

启动脚本：/system/bin

jar包：/system/framework/Monkey.jar

IMG_256

5.2 强制关闭monkey

5.2.1 adb shell ps 查看全部在运行的进程

5.2.2查找出com.android.commands.monkey的进程PID

5.2.3 adb shell kill pid 杀掉monkey进程

5.3 monkey命令

monkey命令格式： adb shell monkey [options] <event-count>

·[options] Monkey可传入的参数，可选，默认无反馈模式，将事件任意发送到安装在目标环境中的全部包

· <event-count> 随机发送的事件数，必选项

5.4 monkey事件

5.4.1 触摸事件 （点击）

5.4.2 手势事件 （滑动）

5.4.3 二指缩放事件

5.4.4 轨迹事件 （一个或多个随机的移动）

5.4.5 屏幕旋转事件

5.4.6 基本导航事件 （上下左右按键 较少）

5.4.7 主要导航事件 （键盘的中间键 回退按键 菜单按键）

5.4.8 系统按键事件 (Home,返回，音量调节...)

5.4.9 启动Activity事件

5.4.10 键盘事件（点击输入框，键盘弹起，点击输入框以外区域，键盘收回）

5.4.11 其他

5.5 monkey参数

5.5.1 常规类参数

1.帮助类参数 monkey -h

2.日志级别

adb shell monkey -v -v -v 1000

1个-v：0级，除启动提示，测试完成和最终结果外提供较少信息

2个-v：1级，提供较详细测试信息，如逐个发送到Activity的时间

3个-v：2级，提供更详细安装信息，如测试中被选中或未被选中的Activity

5.5.2 事件类参数

对随机事件进行调控。具体包括设置各种时间的百分比、设置事件生

成所使用的的种子值等等。频率参数主要限制实践之星的时间间隔

1. 执行指定脚本

adb shell monkey -f <scriptfile> <event-count>

1. 伪随机数生成种子值

使用-s命令可以重复执行之前的伪随机操作。本身每次执行伪随机事件操作也会默认生成一个seed值

adb shell moneky -s <seed> <event-count>

1. 设置间隔 在每个指令之间加上固定的间隔时间

adb shell monkey --throttle <milliseconds> <event-count>

如果throttle后面不跟时间值，默认无间隔

1. 调整触摸事件百分比

adb shell monkey --pct-touch <percentage> <event-count>

1. 调整手势事件百分比

adb shell monkey --pct-motion <percentage> <event-count>

1. 调整应用启动时间百分比

adb shell monkey --pct-appswitch <percentage> <event-count>

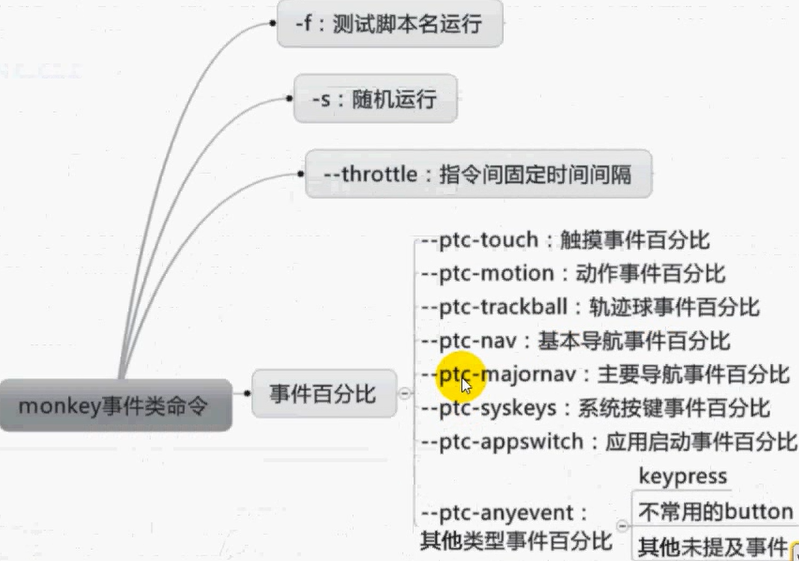
1. 调整屏幕旋转事件百分比

adb shell monkey --pct-rotation <percentage> <event-count>

1. 调整两指缩放时间百分比

adb shell monkey --pct-pinchzoom <percentage> <event-count>

汇总整理：



5.5.3 约束类参数

1.包约束

-p后接一个/多个包名，如果应用需要访问其他包里的activity，则相关的包也需要在此同时指定，如果不指定，将允许系统启动全部包里的activity

adb shell monkey -p <allowed-package-name> <event-count>

2.activity类约束

把monkey限制在一个或几个类别中

Adb shell monkey -c <main-category> <event-count>

5.5.4 调试类参数

1.应用程序崩溃后继续发送时间

adb shell monkey --ignore-crashes <event-count>

2.超时错误继续发送事件

adb shell monkey --ignore-timeouts

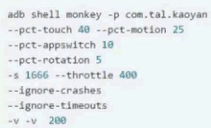
3.应用程序权限发生错误发生后继续发送事件

adb shell monkey --ignore-security-exception

4.其他



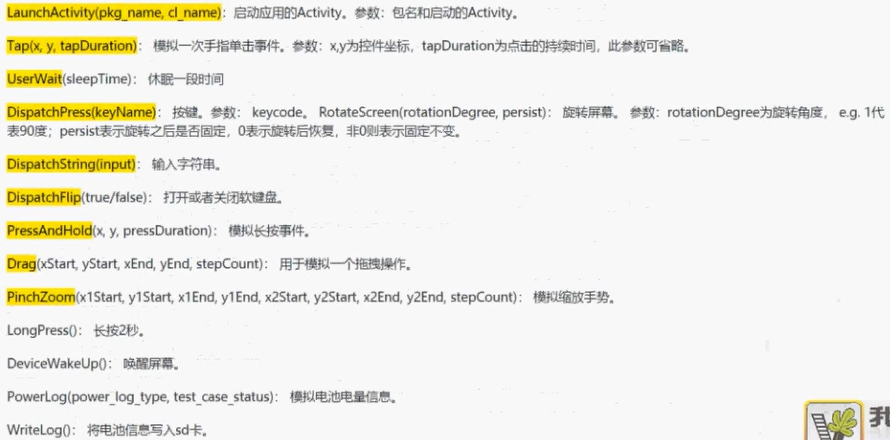
综合测试命令举例：



参考：《深入理解Android自动化测试》

5.6 自定义脚本的稳定性测试

5.6.1 Monkey脚本常用API简介



5.6.2 Monkey脚本格式

Monkey脚本主要包含头文件信息和具体monkey命令两部分

格式：

/\*等号两边要有空格\*/

type = raw events

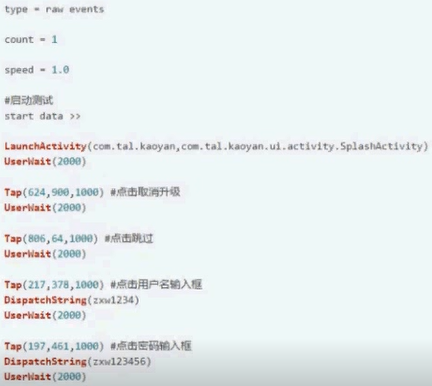
count = 1

Speed = 1.0

Start data >> //启动测试

/\*具体脚本内容 模拟用户操作\*/

Eg.



执行脚本； adb push <remote路径> <local路径>

Adb shell monkey -f <local路径> <event-count>

上述脚本模拟了一个登录过程

5.6.3 monkey日志管理

用于获取当前测试对象在测试过程中是否发生异常及发生的概率、错误信息等

[参考资料]

[Appium安装]

1.<https://www.cnblogs.com/fnng/p/4540731.html>

2.<https://blog.csdn.net/u010381752/article/details/81874273>

3.<http://www.testclass.net/appium/appium-base-sdk>

[自动化测试]

1.<https://baijiahao.baidu.com/s?id=1662021502495073716&wfr=spider&for=pc>

2.链接：IMG_256https://pan.baidu.com/s/1qCBLFZhO8dNCLo7ftoa1MA   
提取码：m2fm